UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/728,441	12/01/2000	Wen-mei W. Hwu	042302 0269900	3226

7590 11/05/2003

PILLSBURY WINTHROP LLP
1600 TYSONS BOULEVARD
INTELLECTUAL PROPERTY DEPARTMENT
MCLEAN, VA 22102

EXAMINER	
LI, AIMEE J	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/05/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/728,441	HWU ET AL.	
	Examiner	Art Unit	
	Aimee J Li	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 December 2000 and 24 May 2002* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-50* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-50* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Claims 1-50 have been considered.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Declaration and corrected papers as received on 30 March 2001 and Change of Address as received on 24 May 2002.

Information Disclosure Statement

3. The listing of references in the specification is not a proper information disclosure statement. 37 CFR 1.98(b) requires a list of all patents, publications, or other information submitted for consideration by the Office, and MPEP § 609 A(1) states, "the list may not be incorporated into the specification but must be submitted in a separate paper." Therefore, unless the references have been cited by the examiner on form PTO-892, they have not been considered. The non-patent literature referred to throughout the specification must be disclosed in an IDS and supplied to the office unless they have been cited by the examiner on form PTO-892.

Priority

4. If applicant desires priority under 35 U.S.C. 119 based upon a previously filed application, specific reference to the earlier filed application must be made in the instant application. For benefit claims under 35 U.S.C. 120, 121 or 365(c), the reference must include the relationship (i.e., continuation, divisional, or continuation-in-part) of the applications. This should appear as the first sentence of the specification following the title, preferably as a separate paragraph unless it appears in an application data sheet. The status of nonprovisional parent

application(s) (whether patented or abandoned) should also be included. If a parent application has become a patent, the expression "now Patent No. _____" should follow the filing date of the parent application. If a parent application has become abandoned, the expression "now abandoned" should follow the filing date of the parent application.

5. If the application is a utility or plant application filed under 35 U.S.C. 111(a) on or after November 29, 2000, the specific reference must be submitted during the pendency of the application and within the later of four months from the actual filing date of the application or sixteen months from the filing date of the prior application. If the application is a utility or plant application which entered the national stage from an international application filed on or after November 29, 2000, after compliance with 35 U.S.C. 371, the specific reference must be submitted during the pendency of the application and within the later of four months from the date on which the national stage commenced under 35 U.S.C. 371(b) or (f) or sixteen months from the filing date of the prior application. See 37 CFR 1.78(a)(2)(ii) and (a)(5)(ii). This time period is not extendable and a failure to submit the reference required by 35 U.S.C. 119(e) and/or 120, where applicable, within this time period is considered a waiver of any benefit of such prior application(s) under 35 U.S.C. 119(e), 120, 121 and 365(c). A priority claim filed after the required time period may be accepted if it is accompanied by a grantable petition to accept an unintentionally delayed claim for priority under 35 U.S.C. 119(e), 120, 121 and 365(c). The petition must be accompanied by (1) the reference required by 35 U.S.C. 120 or 119(e) and 37 CFR 1.78(a)(2) or (a)(5) to the prior application (unless previously submitted), (2) a surcharge under 37 CFR 1.17(t), and (3) a statement that the entire delay between the date the claim was due under 37 CFR 1.78(a)(2) or (a)(5) and the date the claim was filed was unintentional. The

Director may require additional information where there is a question whether the delay was unintentional. The petition should be addressed to: Mail Stop Petition, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Specification

6. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Objections

7. Claims 19-21 are objected to because of the following informalities: Please insert "modulo variable expansion (MVE)" into the claims to ensure there is no discrepancy on the type of code is being referred to. Appropriate correction is required.

8. Claim 38 objected to under 37 CFR 1.75(c) as being in improper form because a dependent claim refers to a future claim, in this case claim 39. See MPEP § 608.01(n).

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 1-3 are rejected under 35 U.S.C. 102(e) as being taught by Fleck et al., U.S. Patent Number 6,085,315 (herein referred to as Fleck).

11. Referring to claim 1, Fleck has taught a processor comprising:

a. A functional unit adapted to execute an instruction issued to it from a dispatch stage (Fleck column 1, line 30 to column 2, line 2; column 2, lines 26-44; and Figure 1); and

b. A buffer in the dispatch stage coupled to the functional unit adapted to store a plurality of the instructions before issue to the functional unit (Fleck column 1, line 30 to column 2, line 2; column 2, lines 45-67; and Figure 1).

12. Referring to claim 2, Fleck has taught a processor according to claim 1, further comprising a decode stage register coupled between the dispatch stage and the functional unit, the functional unit coupled to the decode stage register for executing the instruction issued to the decode stage register from the dispatch stage (Fleck column 2, lines 35-55; column 3, lines 19-31; and Figure 1).

13. Referring to claim 3, Fleck has taught a processor according to claim 1, further comprising control logic coupled to the buffer adapted to cause a certain one of the stored plurality o instructions to be issued to the functional unit in accordance with a loop iteration stage (Fleck column 2, lines 45-67; column 4, lines 9-56; Figure 1; and Figure 3).

Claim Rejections - 35 USC § 103

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

15. Claims 4-18, 26-33, 35-41, and 43-49 rejected under 35 U.S.C. 103(a) as being unpatentable over Fleck et al., U.S. Patent Number 6,085,315 (herein referred to as Fleck) in view of Subramanian et al., U.S. Patent Number 5,867,711 (herein referred to as Subramanian).

16. Referring to claims 4-14, Fleck has taught

a. Control logic coupled to the buffer (Applicant's claim 8) (Fleck column 2, lines 45-67; column 4, lines 9-56; Figure 1; and Figure 3);

b. The control logic including:

i. A loop iteration register for storing a loop iteration parameter (Applicant's claims 8, 10, 12, and 14) (Fleck column 3, line 48 to column 4, line 56 and Figure 3), and

ii. A loop cycles register for storing a loop cycles parameter (Applicant's claims 8, 10, 12, and 14) (Fleck column 3, line 48 to column 4, line 56 and Figure 3);

c. Wherein the control logic is adapted to cause the plurality of instructions to be issued to the functional unit from the buffer in accordance with the loop iteration parameter and the loop cycles parameter (Applicant's claim 8) (Fleck column 3, line 48 to column 4, line 56 and Figure 3); and

d. Control logic coupled to the buffer, the control logic being operative so that the functional unit executes a number of loop iterations (Applicant's claims 9, 11, and 13) (Fleck column 1, line 55 to column 2, line 2; column 3, line 48 to column 4, line 56; Figure 1; and Figure 3).

17. Fleck has not taught:

a. Wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with a cycle within the loop iteration stage (Applicant's claim 4);

b. Wherein the buffer is further adapted to store a plurality of loop stage bit masks respectively associated with the plurality of instructions (Applicant's claim 5);

c. Wherein the buffer is further adapted to store a plurality of loop stage bit masks respectively associated with the plurality of instructions (Applicant's claims 6 and 7);

d. Wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with the respective one of the loop stage bit masks associated with the certain one of the instructions (Applicant's claims 6 and 7);

e. The control logic including an iteration initiation register for storing a loop iteration initiation parameter (Applicant's claims 8, 10, 12 and 14); and

f. Wherein the stored plurality of instructions comprises a kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions in accordance with the kernel set of loop instructions and received loop parameters (Applicant's claims 9, 11, and 13).

18. Subramanian has taught:

a. Wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with a cycle within the loop iteration stage

(Applicant's claim 4) (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figures 4; and Figure 5).

b. Wherein the buffer is further adapted to store a plurality of loop stage bit masks respectively associated with the plurality of instructions (Applicant's claim 5) (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5). In regards to Subramanian, there is a time-stamp, which functions similarly to the stage bit masks, assigned to each instruction to denote when an instruction is to be executed.

c. Wherein the buffer is further adapted to store a plurality of loop stage bit masks respectively associated with the plurality of instructions (Applicant's claims 6 and 7) (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5).

d. Wherein the control logic is further adapted to cause the certain one of the instructions to be issued in accordance with the respective one of the loop stage bit masks associated with the certain one of the instructions (Applicant's claims 6 and 7) (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5). In regards to Subramanian, there is a time-stamp, which functions similarly to the stage bit masks, assigned to each instruction to denote when an instruction is to be executed.

e. The control logic including an iteration initiation register for storing a loop iteration initiation parameter (Applicant's claims 8, 10, 12 and 14) (Subramanian column 5, lines 49-56 and Figure 5).

f. Wherein the stored plurality of instructions comprises a kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions in accordance with the kernel set of loop instructions and received loop parameters (Applicant's claims 9, 11, and 13) (Subramanian column 6, lines 4-19 and Figure 5).

19. In regards to Subramanian, the elements of the claims have been taught by Subramanian as necessary parts of modulo scheduling. A person of ordinary skill in the art at the time the invention was made would have recognized that modulo scheduling is a form of software pipelining specifically for loops, so that successive execution iterations are overlapped (Subramanian column 2, lines 8-10), thereby increasing processor efficiency and speed by executing multiple instructions at the same time. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the modulo scheduling of Subramanian in the device of Fleck to increase processor efficiency and speed.

20. Referring to claims 15 and 16, Fleck has taught a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the stored plurality of instructions (Fleck column 1, line 50 to column 2, line 2; column 3, line 48 to column 4, line 56; Figure 1; and Figure 3).

21. Referring to claims 17 and 18, Fleck has taught a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the loop instructions (Fleck column 1, line 50 to column 2, line 2; column 3, line 48 to column 4, line 56; Figure 1; and Figure 3). Fleck has not taught prologue, kernel, and epilogue sets of loop instructions.

Subramanian has taught prologue, kernel, and epilogue sets of loop instructions (Subramanian column 6, lines 4-19 and Figure 5). In regards to Subramanian, the prologue, kernel, and epilogue sets of loop instructions are identified as part of the necessary breakdown of a loop for modulo scheduling. A person of ordinary skill in the art at the time the invention was made would have recognized that modulo scheduling is a form of software pipelining specifically for loops, so that successive execution iterations are overlapped (Subramanian column 2, lines 8-10), thereby increasing processor efficiency and speed by executing multiple instructions at the same time. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the modulo scheduling of Subramanian in the device of Fleck to increase processor efficiency and speed.

22. Referring to claim 26, Fleck has taught a buffer in the dispatch stage of a processor, the buffer being associated with a functional unit that executes instructions issued to it from the dispatch stage (Fleck column 1, lines 30-2; column 2, lines 26-67; Figure 1; and Figure 3). Fleck has not taught:

a. A kernel set of loop instructions;

b. A plurality of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions.

23. However, Fleck has taught the buffer contains loop instructions (Fleck column 1, lines 30-2; column 2, lines 26-67; Figure 1; and Figure 3). Subramanian has taught:

a. A kernel set of loop instructions (Subramanian column 5, line 29 to column 6, line 15 and Figure 5);

b. A plurality of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5).

24. In regards to Subramanian, the elements of the claim have been taught by Subramanian as necessary parts of modulo scheduling. A person of ordinary skill in the art at the time the invention was made would have recognized that modulo scheduling is a form of software pipelining specifically for loops, so that successive execution iterations are overlapped (Subramanian column 2, lines 8-10), thereby increasing processor efficiency and speed by executing multiple instructions at the same time. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the modulo scheduling of Subramanian in the device of Fleck to increase processor efficiency and speed.

25. Referring to claims 27-31, Fleck has taught

a. Wherein the processor further includes control logic, the buffer being coupled to the control logic and the functional unit for causing the kernel set of loop instructions to be issued to the functional unit (Applicant's claim 27) (Fleck column 1, line 30 to column 2, line 2; column 2, lines 45-67; and Figure 1);

b. Wherein the loop instructions comprise undecoded instructions, and wherein the processor further includes a decode stage interposed between the functional unit and the buffer for decoding the instructions (Applicant's claim 30) (Fleck column 2, lines 35-55; column 3, lines 19-31; and Figure 1); and

c. Wherein the loop instructions comprise decoded instructions in the form of functional unit control signals (Applicant's claim 31) (Fleck column 2, lines 35-

55; column 3, lines 19-31; and Figure 1). In regards to Fleck, instructions are functional unit control signals since they control how the functional unit operates.

26. Fleck has not taught

a. The kernel set of loop instructions and modulo schedule stage identifiers (Applicant's claim 27);

b. Wherein the modulo schedule stage identifiers comprise bit fields (Applicant's claim 28);

c. The control logic determining whether to issue a certain one of the loop instructions to the functional unit in accordance with a bit position of a set bit in the bit field corresponding to the certain loop instruction (Applicant's claim 28); and

d. Wherein the control logic is operative to cause a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions in accordance with the stored modulo schedule stage identifiers (Applicant's claim 29).

27. Subramanian has taught:

a. The kernel set of loop instructions and modulo schedule stage identifiers (Applicant's claim 27) (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5);

b. Wherein the modulo schedule stage identifiers comprise bit fields (Applicant's claim 28) (Subramanian column 3, lines 36-44);

c. The control logic determining whether to issue a certain one of the loop instructions to the functional unit in accordance with a bit position of a set bit in the bit field corresponding to the certain loop instruction (Applicant's claim 28) (Subramanian column 2, lines 10-16; column 4, lines 16-26; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5); and

d. Wherein the control logic is operative to cause a number of loop iterations of the kernel set of loop instructions, a prologue set of loop instructions different than the kernel set of loop instructions, and an epilogue set of loop instructions different than the kernel set of loop instructions in accordance with the stored modulo schedule stage identifiers (Applicant's claim 29) (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 6, lines 4-19; column 10, lines 5-9; Figure 4; and Figure 5).

28. In regards to Subramanian, the elements of the claims have been taught by Subramanian as necessary parts of modulo scheduling. A person of ordinary skill in the art at the time the invention was made would have recognized that modulo scheduling is a form of software pipelining specifically for loops, so that successive execution iterations are overlapped (Subramanian column 2, lines 8-10), thereby increasing processor efficiency and speed by executing multiple instructions at the same time. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the modulo scheduling of Subramanian in the device of Fleck to increase processor efficiency and speed.

29. Referring to claim 32, Fleck has taught a processor for executing a number of iterations of a loop comprising:

a. A plurality of functional units (Fleck column 1, line 30 to column 2, line 2; column 2, lines 26-44; and Figure 1).

b. A dispatch stage coupled to the functional units for issuing instructions to the functional units (Fleck column 1, line 30 to column 2, line 2; column 2, lines 45-67; and Figure 1).

c. A plurality of buffers adapted to store loop instructions (Fleck column 1, line 30 to column 2, line 2; column 2, lines 45-67; and Figure 1). In regards to Fleck, a buffer to store a loop instruction has been taught and duplicating this part is not a patentable improvement. See I*n re Harza* 274 F.2d 669, 124 USPQ 378 (CCPA 1960).

d. Control logic coupled to the plurality of buffers for causing the stored kernel set of instructions to be selectively issued to the functional units (Fleck column 2, lines 45-67; column 4, lines 9-56; Figure 1; and Figure 3).

30. Fleck has not taught

a. The loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions; and

b. The control logic being operative so that the functional units execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored kernel set of loop instructions.

31. Subramanian has taught

a. The loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions (Subramanian column 6, lines 4-19 and Figure 5); and

b. The control logic being operative so that the functional units execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored kernel set of loop instructions (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5).

32. In regards to Subramanian, the elements of the claim have been taught by Subramanian as necessary parts of modulo scheduling. A person of ordinary skill in the art at the time the invention was made would have recognized that modulo scheduling is a form of software pipelining specifically for loops, so that successive execution iterations are overlapped (Subramanian column 2, lines 8-10), thereby increasing processor efficiency and speed by executing multiple instructions at the same time. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the modulo scheduling of Subramanian in the device of Fleck to increase processor efficiency and speed.

33. Referring to claim 33, Fleck has taught a fetch unit, wherein the control logic is further operative so that the fetch unit can be shut down during execution of the number of iterations of the loop (Fleck column 1, line 50 to column 2, line 2; column 3, line 48 to column 4, line 56; Figure 1; and Figure 3).

34. Referring to claims 35 and 43, Fleck has taught a method for executing a number of iterations of a loop in a processor, the method comprising:

a. Storing the kernel set of loop instructions at a dispatch stage of the processor (Fleck column 1, line 30 to column 2, line 2; column 2, lines 45-67; and Figure 1); and

b. Storing loop parameters in control logic associated with the stored loop instructions (Fleck column 2, lines 45-67; column 4, lines 9-56; Figure 1; and Figure 3).

35. Fleck has not taught:

a. The loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions; and

b. Causing the stored kernel set of instructions to be selectively issued to functional units of the processor in accordance with the stored loop parameters so that the functional units of the processor execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of loop instructions based on the stored loop instructions.

36. Subramanian has taught:

a. The loop including a prologue set of loop instructions, a kernel set of loop instructions and an epilogue set of loop instructions (Subramanian column 6, lines 4-19 and Figure 5); and

b. Causing the stored kernel set of instructions to be selectively issued to functional units of the processor in accordance with the stored loop parameters so that the functional units of the processor execute the number of iterations of the kernel set of loop instructions, the prologue set of loop instructions and the epilogue set of

loop instructions based on the stored loop instructions (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5).

37. In regards to Subramanian, the elements of the claim have been taught by Subramanian as necessary parts of modulo scheduling. A person of ordinary skill in the art at the time the invention was made would have recognized that modulo scheduling is a form of software pipelining specifically for loops, so that successive execution iterations are overlapped (Subramanian column 2, lines 8-10), thereby increasing processor efficiency and speed by executing multiple instructions at the same time. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the modulo scheduling of Subramanian in the device of Fleck to increase processor efficiency and speed.

38. Referring to claims 36-40 and 44-48, Fleck has taught

a. A loop iteration register for storing a loop iteration parameter (Applicant's claims 36 and 44) (Fleck column 3, line 48 to column 4, line 56 and Figure 3), and

b. A loop cycles register for storing a loop cycles parameter (Applicant's claims 36 and 44) (Fleck column 3, line 48 to column 4, line 56 and Figure 3);

39. Fleck has not taught:

a. The control logic including an iteration initiation register for storing a loop iteration initiation parameter (Applicant's claims 36 and 44);

b. Adjusting the value in the loop cycles register from the stored loop cycles parameter in accordance with a processor cycle (Applicant's claims 37 and 45);

c. Resetting the value in the loop cycles register in accordance with the adjusted value and the stored iteration initiation parameter (Applicant's claims 37 and 45);

d. Adjusting the value in the loop iteration register in accordance with the resetting step (Applicant's claims 38 and 46);

e. Completing the issue of loop instructions in accordance with the adjusted value in the loop iteration register (Applicant's claims 38 and 46);

f. Storing a set of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions, the step of causing the stored kernel set of loop instructions to be selectively issued including the step of comparing the stored loop parameters with the modulo schedule stage identifiers (Applicant's claims 39 and 47); and

g. Wherein the step of comparing the stored loop parameters with the modulo schedule stage identifiers includes the step of indexing to a certain one of the modulo schedule stage identifiers in accordance with a current cycle in the modulo schedule stage indicated by the stored loop parameters, the step of causing the stored kernel set of loop instructions to be selectively issued further including issuing the associated loop instruction to the functional unit if the certain modulo schedule stage identifier indicates that the functional unit is active (Applicant's claims 40 and 48).

40. Subramanian has taught:

a. The control logic including an iteration initiation register for storing a loop iteration initiation parameter (Applicant's claims 36 and 44) (Subramanian column 5, lines 49-56 and Figure 5);

b. Adjusting the value in the loop cycles register from the stored loop cycles parameter in accordance with a processor cycle (Applicant's claims 37 and 45) (Subramanian column 5, line 29 to column 7, line 7; Figure 4; and Figure 5);

c. Resetting the value in the loop cycles register in accordance with the adjusted value and the stored iteration initiation parameter (Applicant's claims 37 and 45) (Subramanian column 5, line 29 to column 7, line 7; Figure 4; and Figure 5);

d. Adjusting the value in the loop iteration register in accordance with the resetting step (Applicant's claims 38 and 46) (Subramanian column 5, line 29 to column 7, line 7; Figure 4; and Figure 5);

e. Completing the issue of loop instructions in accordance with the adjusted value in the loop iteration register (Applicant's claims 38 and 46) (Subramanian column 5, line 29 to column 7, line 7; Figure 4; and Figure 5);

f. Storing a set of modulo schedule stage identifiers respectively associated with the kernel set of loop instructions, the step of causing the stored kernel set of loop instructions to be selectively issued including the step of comparing the stored loop parameters with the modulo schedule stage identifiers (Applicant's claims 39 and 47) (Subramanian column 2, lines 10-16; column 5, lines 29-45; column 10, lines 5-9; Figure 4; and Figure 5); and

g. Wherein the step of comparing the stored loop parameters with the modulo schedule stage identifiers includes the step of indexing to a certain one of the modulo schedule stage identifiers in accordance with a current cycle in the modulo schedule stage indicated by the stored loop parameters, the step of

causing the stored kernel set of loop instructions to be selectively issued further including issuing the associated loop instruction to the functional unit if the certain modulo schedule stage identifier indicates that the functional unit is active (Applicant's claims 40 and 48) (Subramanian column 5, line 29 to column 7, line 7; Figure 4; and Figure 5).

41. In regards to Subramanian, the elements of the claims have been taught by Subramanian as necessary parts of modulo scheduling. A person of ordinary skill in the art at the time the invention was made would have recognized that modulo scheduling is a form of software pipelining specifically for loops, so that successive execution iterations are overlapped (Subramanian column 2, lines 8-10), thereby increasing processor efficiency and speed by executing multiple instructions at the same time. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the modulo scheduling of Subramanian in the device of Fleck to increase processor efficiency and speed.

42. Referring to claims 41 and 49, Fleck has taught shutting down the fetch unit during execution of the number of iterations of the loop (Fleck column 1, line 50 to column 2, line 2; column 3, line 48 to column 4, line 56; Figure 1; and Figure 3).

43. Claims 19-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fleck in view of Subramanian as applied to claims 9, 11, and 13 above, and further in view of Valluri and Govindarajan's "Modulo-Variable Expansion Sensitive Scheduling" published in High Performance Computing, 1998 (herein referred to as Valluri). Fleck in view of Subramanian has not taught wherein the kernel set of loop instructions comprise MVE code. Valluri has taught wherein the kernel set of loop instructions comprise MVE code (Valluri section 1. Introduction,

paragraphs 1-2). A person of ordinary skill in the art at the time the invention was made would have recognized that MVE is needed to handle overlapping of a single variable with a subsequent definition of itself by ensuring that different registers are used each time (Valluri section 1. Introduction, paragraphs 1-2). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate MVE of Valluri in the device of Fleck in view of Subramanian to handle same variable overlap.

44. Claims 22-25, 34, 42, and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fleck in view of Subramanian as applied to claims 3, 8, 11, and 13 above, and further in view of Morrison et al., U.S. Patent Number 6,421,744 (herein referred to as Morrison). Fleck has not taught wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration. Morrison has taught wherein the control logic is further operative to allow interrupts to be handled at the end of a current loop iteration and to complete the number of loop iterations after the interrupt is handled (Morrison column 9, lines 18-23). In regards to Morrison, returning to complete the loop iterations is inherent to interrupts, since they are only temporary. Please see InstantWeb's Online Computing Dictionary. A person of ordinary skill in the art at the time the invention was made would have recognized that waiting until the end of a loop iteration to allow interrupts would ensure data is not lost and/or corrupted and continuing afterwards ensures the process completes and normal process has resumed. Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the interrupts of Morrison in Fleck.

Conclusion

45. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as follows. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

a. Tirumalai et al., U.S. Patent Number 5,835,774, has taught modulo-scheduled loops.

b. Tirumalai, U.S. Patent Number 5,809,308, has taught modulo-scheduled loops.

c. Tirumalai, U.S. Patent Number 5,664,193, has taught modulo-scheduled loops.

d. Kiuchi et al., U.S. Patent Number 5,579,493, has taught a loop buffer.

e. Ganapathy et al., U.S. Patent Number 6,598,155, has taught a loop execution device with loop buffer.

f. Fernando et al., U.S. patent Number 6,269,440, has taught a loop execution device with loop buffer.

g. Cao et al., U.S. Patent Number 6,361,071, has taught a loop buffer.

h. H. Fatih Uğurdağ and Christos A. Papachristou's "A VLIW Architecture Based on Shifting Register Files" has taught modulo-scheduling and modulo-variable expansion.

i. Nancy J. Warter, Daniel M. Lavery, and Wen-mei W. Hwu's "The Benefit of Predicated Execution for Software Pipelining" has taught loop handling and modulo scheduling.

j. P. Tirumalai, M. Lee, and M. Schlansker's "Parallelization of Loops with Exits on Pipelined Architectures" has taught loop handling and modulo scheduling.

k. Nancy J. Warter, Grant E. Haab, Krishna Subramanian, and John W. Bockhaus's "Enhanced Modulo Scheduling for Loops with Conditional Branches" has taught loop handling and modulo scheduling.

l. Daniel M. Lavery and Wen-mei W. Hwu's "Modulo Scheduling of Loops in Control-Intensive Non-Numeric Programs" has taught loop handling and modulo scheduling.

m. Uma Mahadevan, Kevin Nomura, Roy Dz-ching Ju, and Rick Hank's "Applying Data Speculation in Modulo Scheduled Loops" has taught loop handling and modulo scheduling.

46. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J Li whose telephone number is (703) 305-7596. The examiner can normally be reached on M-T 7:30am-5:00pm.

47. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703) 305-9712. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

48. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Aimee J. Li
Examiner
Art Unit 2183

November 3, 2003

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100